Encl (1)

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) FINDING PARITY IN A BROADCAST NETWORK		5. TYPE OF REPORT & PERIOD COVERED paper 6. PERFORMING ORG. REPORT NUMBER LIDS-P-1490
7. AUTHOR(s) Robert G. Gallager		8. CONTRACT OR GRANT NUMBER(s) DARPA Order No. 3045/2-2-84 Amendment #11 ONR/N00014-84-K-0357
9. PERFORMING ORGANIZATION NAME AND ADDRESS Massachusetts Institute of Technology Laboratory for Information and Decision Systems Cambridge, Massachusetts 02139		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Program Code No. 5T10 ONR Identifying No. 049-383
11. CONTROLLING OFFICE NAME AND ADDRESS Defense Advanced Research Projects Agency 1400 Wilson Boulevard Arlington, Virginia 22209		12. REPORT DATE August 1985 13. NUMBER OF PAGES 11
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Office of Naval Research Information Systems Program Code 437 Arlington, Virginia 22217		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release: distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

DTIC ELECTE SEP 04 1985 S D E

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Consider a broadcast network of N nodes in which each binary digit transmitted by each node is received by each other node via a binary symmetric channel whose crossover probability is independent over transmitters, receivers, and time. Each node has a binary state and the problem is to construct a distributed algorithm to find the parity of the set of states with some given reliability. It is shown that this can be done with O(ln ln N) bits of communication from each node. Communicationg all the node states to one node can be accomplished with only marginally more communication.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

August 1985 LIDS-P-1490

FINDING PARITY IN A BROADCAST NETWORK*

by

Robert G. Gallager**

ABSTRACT

Consider a broadcast network of N nodes in which each binary digit transmitted by each node is received by each other node via a binary symmetric channel whose crossover probability is independent over transmitters, receivers, and time. Each node has a binary state and the problem is to construct a distributed algorithm to find the parity of the set of states with some given reliability. It is shown that this can be done with $0(\ln \ln N)$ bits of communication from each node. Communicating all the node states to one node can be accomplished with only marginally more communication.

---

*This research was conducted at the M.I.T. Laboratory for Information and Decision Systems with partial support provided by the National Science Foundation under Grant NSF-ECS-8510698 and by the Defense Advanced Research Projects Agency under Contract ONR/N00014-84-K-0357.

**Room No. 35-206, Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139

FINDING PARITY IN A BROADCAST NETWORK

by Robert G. Gallager

1) INTRODUCTION

Consider a broadcast network of N+1 nodes in which each binary digit transmitted by each node is received by each other node via a binary symmetric channel whose crossover probability $\epsilon < 1/2$ is independent over transmitters, receivers, and time. Each node has a binary state, and the problem under consideration is for one special node, called the receiver, to determine the parity of the set of node states. In particular, we want to minimize the number of binary digits that must be sent by each node in order for the receiver to determine parity within some allowable error probability, $\bar{P}$. We assume that whatever algorithm is used for transmitting the required information, all nodes know the algorithm and there is no contention between transmissions; thus each binary digit transmitted is received and identified by all other nodes, subject to the noise introduced by the binary symmetric channels.

The above problem was first formulated by A. El Gamal [1], and is of interest because it is one of the simplest distributed algorithm problems involving noise. A closely related problem that we treat is for the receiver to find the state of each other node. Note that the conventional ideas of information theory cannot be used here because each node has only one bit of information to communicate. This situation of communicating a limited amount of information is common in network protocols and more generally in the control of distributed systems. The particular character of the problem here comes from the independence of the noise at each receiver for a given transmission. This means that when a node transmits a single digit, the other nodes collectively could make a good decision on that digit since they have N independent noise samples of it; unfortunately, the nodes cannot act collectively without using up their own valuable transmissions, which are also noisy.

The straightforward approach to this problem is for each node to broadcast its own state j times for some integer j. The receiver will make an error in decoding a given node's state with a probability ϵ_j closely upper bounded [2] by

$$\epsilon_j \leq \alpha^{-j} \quad \text{where} \quad \alpha = [4\epsilon(1-\epsilon)]^{-1/2} \qquad (1)$$

The probability P that the receiver will make an error in calculating the parity of the states is then upper bounded by $N\alpha^{-j}$. Since this bound is quite tight for $N\alpha^{-j}$ small, we see that j must grow as ln(N) in this approach for a constant P.

2) FINDING PARITY WITH O[ln(ln(N))] BITS PER NODE

The approach we take here for more efficient communication is to partition the nodes (other than the receiver) into subsets each with approximately the same number of nodes. In particular, it is always possible to partition N nodes into subsets of k or k-1 nodes each for any k satisfying $(k-1)^2 \leq N$. Each node again broadcasts its own state j times, but then makes a decision on the state of each of the other nodes in its subset using the j receptions from that node. The node adds these decisions modulo 2 to estimate parity for its own subset, and then broadcasts this estimated parity exactly once. The receiver will then receive k or k-1 different estimates for the parity of each subset, and, as we shall see, this allows the receiver to obtain a highly reliable decision on a subset's parity with a relatively small value of j. Given the parity of each subset, the parity of the entire set is found by addition modulo 2 of the individual subset parities.

Note that the parity estimate that the receiver obtains from a given node can be wrong either because of noise in the transmission of the parity or because of decision errors at the sending node. In particular, a given received estimate is incorrect if an odd number of errors occur, counting both the sending node's transmission of parity and its decision on each of the other states in the subset; the probability of this is upper bounded by

$$\beta = [1 - (1-\epsilon_j)^{k-1}(1-2\epsilon)]/2. \qquad (2)$$

This upper bound is met with equality for a reception from a node in a subset of size k, and with inequality for size k-1. β also upper bounds the probability that the receiver's internal estimate of a subset's parity is incorrect. Finally the receiver decides on the parity of a subset by taking a majority vote among the received estimates and its own internal estimate. Since the errors in these estimates are independent, the probability that

half or more of the estimates are erroneous is upper bounded by

$$P_{subset} \leq [4\beta(1-\beta)]^{k/2}. \qquad (3)$$

Combining (2) and (3),

$$P_{subset} \leq [1 - (1-2\epsilon_j)^{2(k-1)}(1-\epsilon)^2]^{k/2} \qquad (4)$$

The receiver next adds the parities of all the subsets (of which there are at most N) and adds its own state, all modulo 2. The probability P that this decision on parity for the entire set is incorrect is upper bounded by the probability of an error on one or more of the subsets, which is further upper bounded by $N P_{subset}$. Thus

$$P \leq N[1 - (1-2\alpha^{-j})^{2(k-1)}(1-2\epsilon)^2]^{k/2} \qquad (5)$$

In going from (4) to (5), we have used (1) to upper bound ϵ_j by α^{-j}. The right hand side of (5) first decreases and then increases with increasing k, and the minimizing integer k is a complicated function of ϵ and j. It is sufficient for our purposes, however, to simply restrict k to be small enough to satisfy

$$(1-2\alpha^{-j})^{2(k-1)} \leq 1/4 \qquad (6)$$

With this restriction,

$$P \leq N Z^{k/2} \qquad \text{where } Z = 1 - (1-2\epsilon)/4 \qquad (7)$$

This is essentially the solution we are looking for; we choose k large enough to make the error probability sufficiently small in (7) and then choose j large enough to satisfy (6). As N is varied, we see that k must increase logarithmically with N, and then j increases logarithmically with k. Since j and k must be integers, however, a little fussing is required to get a valid bound on j. We start by defining real number approximations, $\tilde{k}$ and $\tilde{j}$, to k and j for a given number of nodes N and a given requirement $\tilde{P}$ on error probability:

$$\tilde{k} = [2 \ln(N/\tilde{P})]/\ln Z^{-1}; \quad \tilde{j} = [\ln(2) - \ln(1-2^{-1/\tilde{k}})]/\ln(\alpha) \qquad (8)$$

$$k = \lceil \tilde{k} \rceil; \qquad j = \lceil \tilde{j} \rceil \quad .$$

We have used the notation $\lceil x \rceil$ to mean the smallest integer greater than or equal to x. We now show that with k and j chosen according to (8), the resulting error probability will be at most $\tilde{P}$. Note that the equation for $\tilde{j}$ in (8) can be rearranged to

$$(1-2\alpha^{-\tilde{j}})^{2\tilde{k}} = 1/4 \qquad (9)$$

Since $\tilde{k} > k-1$ and $\tilde{j} \leq j$, (6) must be satisfied. This means that (7) must be satisfied, yielding

$$P \leq N Z^{k/2} \leq N Z^{\tilde{k}/2} = \tilde{P} \qquad (10)$$

The second inequality above is valid because $\tilde{k} \leq k$, and the equality is a rearrangement of the definition of $\tilde{k}$. One final simplification will now be useful in obtaining our final bound. Assume that $\tilde{k}$, as given by (8), satisfies $\tilde{k} \geq 1$. Then it is not hard to verify that $1 - 2^{-1/\tilde{k}} \geq 1/(2\tilde{k})$. Substituting this into the definition of $\tilde{j}$, we obtain

$$\tilde{j} \leq \ln(4\tilde{k}) / \ln(\alpha) \qquad (11)$$

The number of digits transmitted per node is m = j+1, which is at most $\tilde{j}+2$. Substituting this into (11) and using (8) for $\tilde{k}$, we obtain our final bound,

$$m \leq [\ln(\ln(N/\tilde{P})) + A]/\ln(\alpha) + 2 \qquad \text{where} \qquad (12)$$

$$A = \ln(8/\ln(1/Z)) \qquad (13)$$

Recall that we have imposed two restrictions on k in deriving this result. First $N \geq (k-1)^2$, which is satisfied if $N \geq \tilde{k}^2$, and second $\tilde{k} \geq 1$. From the definition of $\tilde{k}$, these restrictions are

$$N \geq [2 \ln(N/\tilde{P})/\ln(1/Z)]^2; \quad N \geq \tilde{P} Z^{-1/2} \qquad (14)$$

The second restriction is always satisfied for N > 1, but the second restriction is more substantive. Note first that (14) is always satisfied for large enough N given any $\tilde{P}$ and ϵ, and thus (12) shows that asymptotically, m increases at most as ln(ln(N)) On the other hand, for given N and ϵ, (14) is always violated for small enough $\tilde{P}$. Thus (12)

(subject to (14)) does not show that m asymptotically varies with $\bar{P}$ as $\ln(\ln(1/\bar{P}))$. This is reassuring, since even if all nodes other than the receiver knew the parity of the states, the error probability could not decrease faster than α^{-Nm}. Actually, by changing the strategy somewhat and making all subsets of size k except for one subset of size between 1 and k, the restriction $N \geq (k-1)^2$ could be relaxed to $N \geq k$. In this case, some of the nodes would have to transmit an extra digit to help resolve the parity of the small subset, and the bound on m would be somewhat weakened. We omit the details of this since it is tedious and doesn't improve the asymptotic behavior with N.

It is also possible to reduce the value of A for large values of ϵ by having each node transmit an estimate of parity for several subsets rather than just its own subset. Again we omit the analysis since it is tedious and does not materially improve the result.

3) FINDING THE STATE OF ALL NODES

In this section we show that the receiver can determine the state of all nodes with very few more transmissions per node than are required to determine parity. Our strategy in doing this is to form a set of N subsets of the N nodes (not counting the receiver) in such a way that each subset contains k-1 or k nodes for some k and each node is contained in k-1 or k subsets. Furthermore, we constrain the choice of subsets so that no pair of nodes appear together in more than one subset. In the appendix, we show that such a set of subsets can always be constructed if

$$N \geq 2k(k-1)^2 \qquad (15)$$

We next associate each node with one subset in a one to one fashion so that each node is associated with a subset containing it and each subset has one of its contained nodes associated with it. The appendix also shows how this association can be constructed. Each node then sends its own state j times and then each node estimates the parity of its associated subset in the same way as before. Finally each node sends the parity of its associated subset and the receiver uses this information, plus its own receptions of the node states, to determine the state of each node. Thus each node sends $m = j+1$ binary digits as before.

Now consider how the receiver can decode the state of each node from the received information. First the receiver makes an internal decision on

the state of each digit from the j noisy receptions of that digit. The probability of error for each of these internal decisions is $\epsilon_j \leq \alpha^{-j}$ as before. In order to make a final decoding of the state of a given node, say node i, the receiver considers each of the subsets, say $S_{i,1}, S_{i,2}, \ldots, S_{i,k}$, or $S_{i,1}, \ldots, S_{i,k-1}$ that contain node i. For a given subset, say $S_{i,\gamma}$, the receiver modulo 2 adds its internal decisions on the nodes in $S_{i,\gamma} - \{i\}$ to the received parity estimate of $S_{i,\gamma}$. Note that if the receivers internal decisions on these nodes are all correct, and if the transmitting nodes decisions on the nodes in the subset are all correct, and if the transmission of that parity is correct, then this modulo 2 sum is simply the state of node i since all other node states are added twice. Thus this sum is in some sense an estimate of the state of node i. More particularly, the probability that this estimate is incorrect is the probability of an odd number of errors in the transmission of parity, in the receiver's internal decisions on the nodes in $S_{i,\gamma}$, and in the transmitting node's decisions on the nodes of the subset other than itself. If the subset contains k nodes, then we are looking at k-1 decisions at the receiver, k-1 at the transmitting node, and one transmission of parity. The probability that this estimate is incorrect is then

$$\beta' = [1 - (1-2\epsilon_j)^{2(k-1)}(1-2\epsilon)]/2 \qquad (16)$$

If the subset contains k-1 nodes, then k-1 would be replaced by k-2 in the above equation, so that β' is an upper bound on the probability of an incorrect estimate in that case. Finally, β' also upper bounds the probability of an error in the receiver's internal estimate of the state of node i. The receiver now has its internal decision of node i's state plus either k or k-1 estimates from the different subsets containing i. Since no two subsets contain more than one node in common, $S_{i,1}-\{i\}$, $S_{i,2}-\{i\}$, ... are all disjoint and all of these estimates are based on mutually independent errors. Thus when the receiver takes a majority vote on its k or k+1 estimates, the probability of error in this final decision on node i is upper bounded by

$$P_i \leq [4\beta'(1-\beta')]^{k/2} \qquad (17)$$

Combining (16) and (17) and upper bounding ϵ_j with α^{-j}, we get

$$P_j \leq [1 - (1-2\alpha^{-j})^{4(k-1)}(1-2\epsilon)^2]^{k/2} \qquad (18)$$

The probability that any node state will be decoded incorrectly is now upper bounded by $P \leq N P_j$. We now restrict k to be small enough that

$$(1-2\alpha^{-j})^{4(k-1)} \leq 1/4 \quad \text{yielding} \qquad (19)$$

$$P \leq N Z^{k/2}; \quad Z = 1 - (1-2\epsilon)^2 \qquad (20)$$

As in the analysis of finding the parity of the states, we now consider a required error probability, $\tilde{P}$, and choose k and j to meet the requirement.

$$\tilde{k} = 2 \ln(N/\tilde{P})/\ln(1/Z); \quad \tilde{j} = [\ln(2) - \ln(1-2^{-1/(2\tilde{k})}]/\ln(\alpha) \qquad (21)$$

$$k = \lceil \tilde{k} \rceil; \qquad j = \lceil \tilde{j} \rceil$$

As before, this guarantees that (19) is satisfied and that the error probability is at most $\tilde{P}$. If $2\tilde{k} \geq 1$, we can upper bound $\tilde{j}$ by $\ln(4\tilde{k})/\ln(\alpha)$. Since the number of digits per node is at most $\tilde{j}+2$, we can use this bound on $\tilde{j}$ with the definition of $\tilde{k}$ to obtain

$$m \leq [\ln(\ln(N/\tilde{P})) + A']/\ln(\alpha) + 2 \quad \text{where} \qquad (22)$$

$$A' = \ln(16/\ln(1/Z)) \qquad (23)$$

Note that the required number of transmission here exceeds that for finding parity only by $\ln(2)/\ln(\alpha)$. Recall that the construction here required $N \geq 2k(k-1)^2$, which is valid for $N \geq 2\tilde{k}^2(\tilde{k}+1)$. From (21), we see that this is satisfied, for any given $\tilde{P}$ and ϵ for all sufficiently large N. As before, for fixed N and ϵ, the restriction is always violated for small enough $\tilde{P}$.

An interesting question raised by these results is whether strategies exist for which m can grow asymptotically even more slowly than as $\ln(\ln(N))$. We conjecture that the answer is no.

APPENDIX

We start with a set of N nodes and want to construct N subsets of k or k-1 nodes each with the property that each node is in k or k-1 subsets and no two subsets contain any pair of nodes in common. Assume that $N \geq 2k(k-1)^2$. Let $L = \lceil N/k \rceil$ and let $N' = Lk$. We first construct N' subsets of k nodes each from a set of N' nodes and then delete N'-N nodes and subsets.

Consider the problem in terms of an N' by N' matrix of 0's and 1's. The columns correspond to nodes and the rows to subsets. The 1's in a row correspond to the nodes in the corresponding subset, so our problem is to construct an N' by N' matrix in which each row and each column contains k 1's and for any two rows, there is at most one column for which both rows contain 1's. We construct such a matrix by first considering the N' by N' matrix as partitioned into k^2 L by L permutation matrices. The fact that each L by L matrix is a permutation matrix guarantees that each row and each column of the entire matrix contains k ones. The figure below shows such a partitioned matrix where L = 3 and k = 3; note that no two rows contain more than a single 1 in common.

1	0	0	1	0	0	1	0	0
0	1	0	0	1	0	0	1	0
0	0	1	0	0	1	0	0	1
1	0	0	0	1	0	0	0	1
0	1	0	0	0	1	1	0	0
0	0	1	1	0	0	0	1	0
1	0	0	0	0	1	0	1	0
0	1	0	1	0	0	0	0	1
0	0	1	0	1	0	1	0	0

Figure 1

Next consider the construction of the L by L permutation submatrices. Suppose we have already selected some of them and are selecting a new one. For a given row, there are at most k-1 1's already in that row (in the

other submatrices), and for each of those 1's, there are at most k-1 other rows that have 1's in the same column. Each of these (at most $(k-1)^2$) rows have at most a single one in the colums of the new permutation submatrix being constructed, and these columns are unavailable for the given row of the permutation matrix; if a 1 was placed in one of these columns in the given row, then that row and another row would have two 1's in common. Using the same argument on columns, we see that each column of the submatrix has at most $(k-1)^2$ row positions unavailable for 1's. The constraint $N \geq 2k(k-1)^2$ implies that $L \geq 2(k-1)^2$, and thus each row and column has at least L/2 positions available for 1's without violating the condition that no two rows have two 1's in commo.

Now suppose we mark all the available positions in the new submatrix and start to construct a permutation matrix using only available positions for 1's; the problem then is to place a single 1 in each row and column, using only available positions. We place 1's into the submatrix one at a time, and after each 1 is placed, the corresponding row and column are considered as blocked. Thus at each step, we look for an available position for which the row and column are unblocked. If the procedure terminates with a 1 in each row and column, the permutation submatrix is complete and we proceed to the next submatrix until all submatrices are complete. Suppose, however, that at some point there is at least one unblocked row and column with no available unblocked positions in either. This row and column each have at least L/2 available but blocked positions. The submatrix at this point contains at most L-1 1's. Thus at least one of the 1's in the submatrix must be blocking both an available position in the given row and an available position in the given column. Thus by removing this single 1, we can add two 1's, one in the given row and another in the given column. By repeating this procedure if need be, the given permutation submatrix can be completely constructed. The figure below illustrates this procedure. An A indicates a position that is available but in a blocked row or column and a U indicates an unavailable position.

The constraint $L \geq (k-1)^2$ that we have imposed appears to be stronger than necessary in many cases. For example, if L is prime and L = k, it is possible to construct a matrix with the required properties as follows: let P_i denote the cyclic L by L permutation matrix that is shifted i places from the identity matrix. Then for the m^{th} row and n^{th} column submatrix, use P_i where $i = (m-1)(n-1)$.

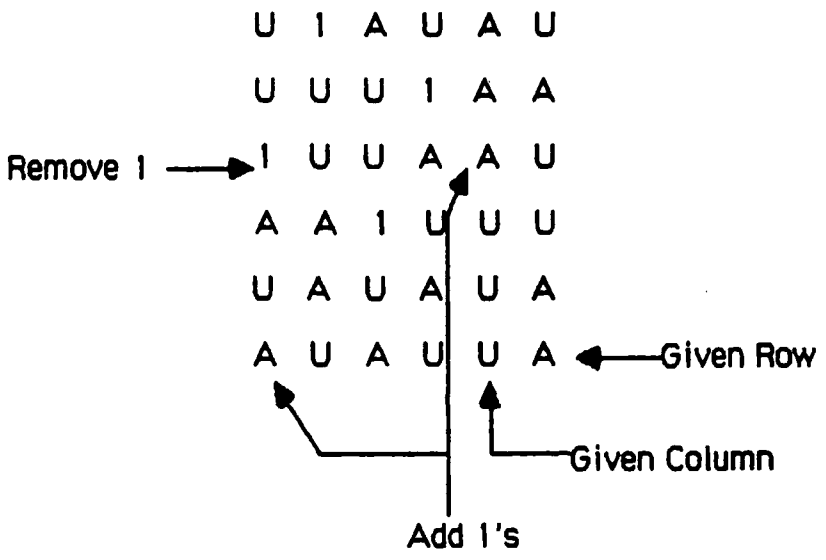

Figure 2

The above paragraphs show how to construct an N' by N' matrix where N' = k L ≥ N. By removing the first N'-N rows and columns, we have an N by N matrix. Since N'-N < L, we have removed at most a single 1 from each row and column, so the remaining matrix satisfies the required properties and has either k or k-1 1's in each row and column. Finally, we can make the top left submatrix be an identity matrix in the original construction. We then associate each column of the N by N matrix with the row for which the main diagonal of submatrices contains a 1. Each row is clearly chosen once and only once by this construction.

REFERENCES

1) El Gamal, A., Open problem presented in the 1984 Workshop on Specific Problems in Communicataion and Computation sponsored by Bell Communication Research.

2) Gallager, R. G., *Information Theory and Reliable Communication*, Section 5.3, John Wiley, N.Y., 1968.

Distribution List

Defense Documentation Center — 12 Copies
Cameron Station
Alexandria, Virginia 22314

Assistant Chief for Technology — 1 Copy
Office of Naval Research, Code 200
Arlington, Virginia 22217

Office of Naval Research — 2 Copies
Information Systems Program
Code 437
Arlington, Virginia 22217

Office of Naval Research — 1 Copy
Branch Office, Boston
495 Summer Street
Boston, Massachusetts 02210

Office of Naval Research — 1 Copy
Branch Office, Chicago
536 South Clark Street
Chicago, Illinois 60605

Office of Naval Research — 1 Copy
Branch Office, Pasadena
1030 East Greet Street
Pasadena, California 91106

Naval Research Laboratory — 6 Copies
Technical Information Division, Code 2627
Washington, D.C. 20375

Dr. A. L. Slafkosky — 1 Copy
Scientific Advisor
Commandant of the Marine Corps (Code RD-1)
Washington, D.C. 20380

Office of Naval Research
Code 455
Arlington, Virginia 22217
1 Copy

Office of Naval Research
Code 458
Arlington, Virginia 22217
1 Copy

Naval Electronics Laboratory Center
Advanced Software Technology Division
Code 5200
San Diego, California 92152
1 Copy

Mr. E. H. Gleissner
Naval Ship Research & Development Center
Computation and Mathematics Department
Bethesda, Maryland 20084
1 Copy

Captain Grace M. Hopper
Naval Data Automation Command
Code 00H
Washington Navy Yard
Washington, DC 20374
1 Copy

Advanced Research Projects Agency
Information Processing Techniques
1400 Wilson Boulevard
Arlington, Virginia 22209
1 Copy

Dr. Stuart L. Brodsky
Office of Naval Research
Code 432
Arlington, Virginia 22217
1 Copy

Prof. Fouad A. Tobagi
Computer Systems Laboratory
Stanford Electronics Laboratories
Department of Electrical Engineering
Stanford University
Stanford, CA 94305